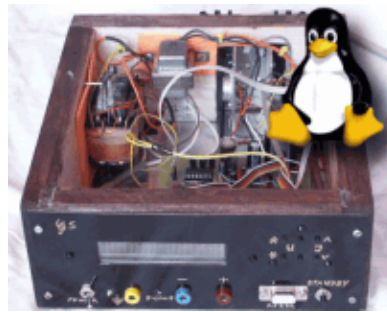


## Alimentation pilotée par micro-contrôleur



par Guido Socher ([homepage](#))

### L'auteur:

Guido aime Linux non seulement parce qu'il est amusant d'en découvrir les grandes possibilités mais aussi à cause des personnes impliquées dans sa conception.

### Traduit en Français par:

Iznogood

<[iznogood/at/iznogood-factory.org](mailto:iznogood/at/iznogood-factory.org)>

### Résumé:

Cet article est le quatrième volet de la série sur les micro-contrôleurs AT90S4433 de [LinuxFocus](#). Je vous suggère de lire les articles précédents sur la programmation des micro-contrôleurs Atmel pour ce qui concerne :

1. Comment installer et utiliser l'environnement de développement d'AVR pour Linux et comment fabriquer le programmeur : [Mars 2002, Programmer le micro-contrôleur AVR avec GCC](#)
2. Comment créer votre propre circuit imprimé : [Mai 2002, Un panneau de contrôle LCD pour votre serveur Linux](#)
3. Comment construire le boîtier de votre alimentation : [Septembre 2002, Compteur de Fréquence 1Hz-100Mhz avec afficheur LCD et interface RS232](#)

Un des éléments les plus importants de votre laboratoire personnel est une bonne alimentation fiable. Dans cet article, nous allons construire cette alimentation. Et elle sera régulée par micro-contrôleur. Elle possède un afficheur LCD et vous pouvez lui envoyer des commandes depuis votre machine Linux par une interface RS232. Elle est également très robuste.

Cet article montre aussi la versatilité des micro-contrôleurs. Ce n'est pas, malgré tout, le circuit le plus simple.

Si vous cherchez une simple alimentation à courant continu, jetez un coup d'oeil à "[une simple alimentation à courant continu](#)". L'alimentation simple est suffisante pour les autres expérimentations électroniques présentées dans [LinuxFocus](#). Cela n'a néanmoins rien à voir avec Linux et les logiciels en général.

Même si vous avez seulement construit l'unité "alimentation simple à courant continu", cette lecture vous permettra de découvrir de nombreux aspects intéressants des micro-contrôleurs.

# Introduction

Cette alimentation pilotée par micro–contrôleur ne possède pas le circuit le plus simple mais je peux vous assurer que vous ne regretterez pas le temps passé à la construire. Elle est très robuste et fiable. Elle est aussi techniquement très intéressante car vous apprendrez comment générer une tension analogique avec un micro–contrôleur sans convertisseur numérique–analogique.

Il vous faut de nombreux composants pour la réalisation de cette alimentation, mais ils sont tous bon marché. Elle ne coûte vraiment pas cher.

## Ce dont vous avez besoin

Lisez la [liste des composants](#) pour savoir ce qu'il vous faut. Vous pouvez aussi voir les composants et leurs valeurs dans le schéma ci–dessous.

Il existe trois variantes de notre alimentation. A l'exception du transformateur et d'une résistance, les différences ne concernent que le logiciel. Tous les autres composants sont identiques pour les 3 options :

1. 0–16V  $I_{max}=2.2A$   
acheter un transformateur de 15V 2.5A
2. 0–24V  $I_{max}=2.2A$   
acheter un transformateur de 24V 2.5A
3. 0–30V  $I_{max}=3A$   
acheter un transformateur de 30V 3A

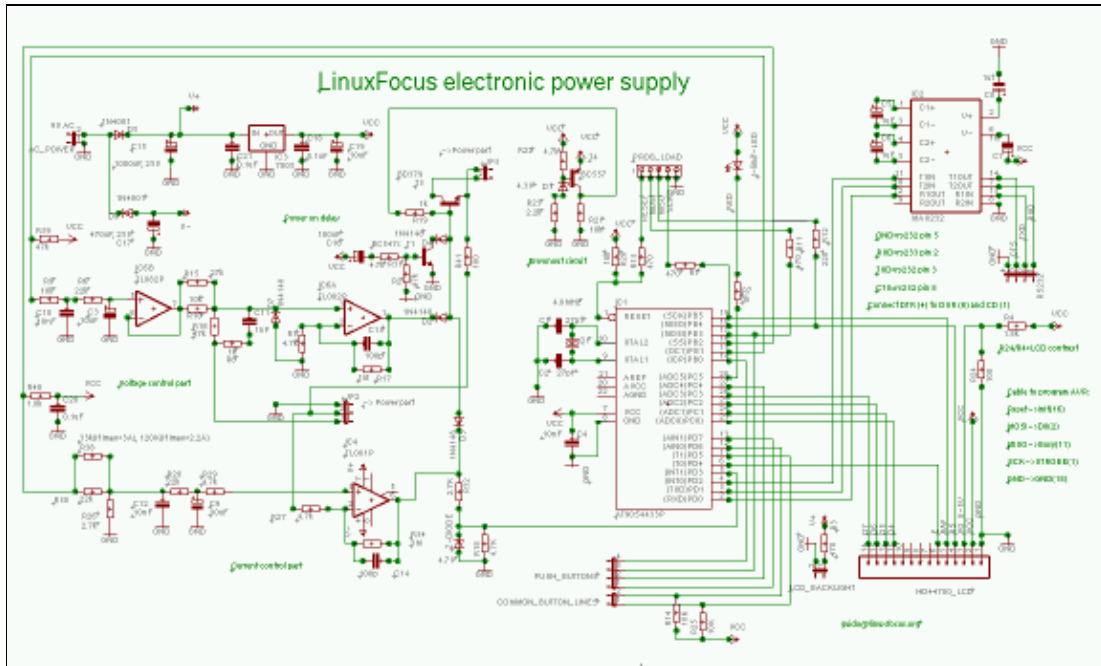
Note : Dans les trois cas, il vous faut évidemment un transformateur additionnel 9V, 100mA pour l'alimentation de la carte principale.

## Schéma et circuit

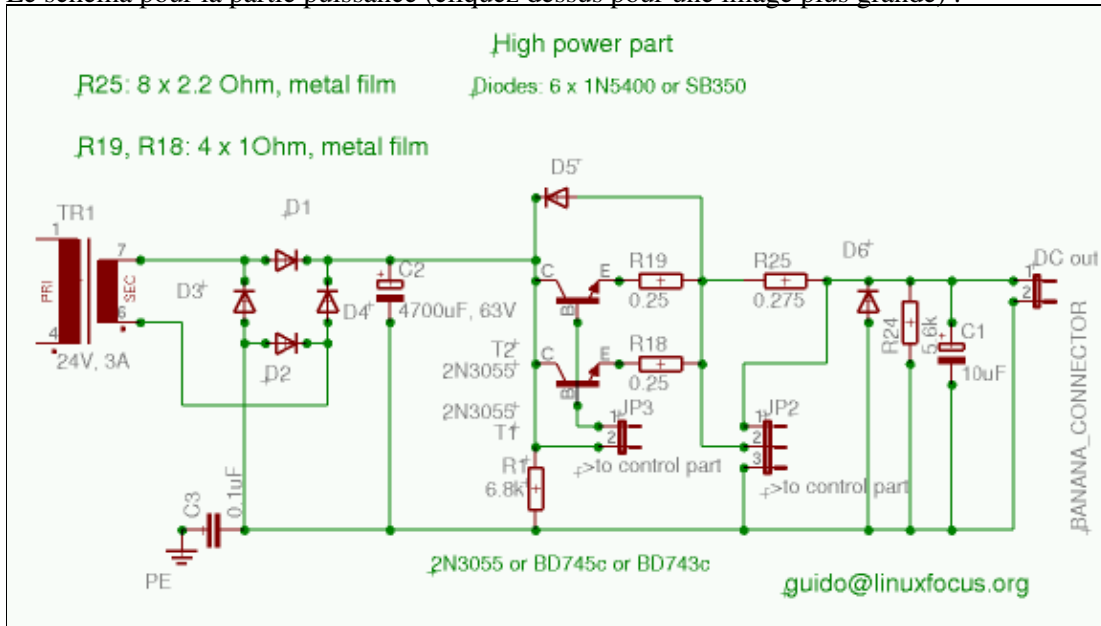
J'ai utilisé [eagle](#) pour Linux pour concevoir le schéma et le circuit. Les fichiers eagle sont aussi inclus dans le paquetage tar.gz avec le logiciel. Vous pouvez le télécharger à la fin de l'article.

Le circuit est divisé en deux parties. Une partie principale et une partie qui doit être à proximité des transistors de puissance. Vous voyez ci–dessous deux schémas indépendants pour les deux blocs mais ils doivent ensuite être connectés par des fils.

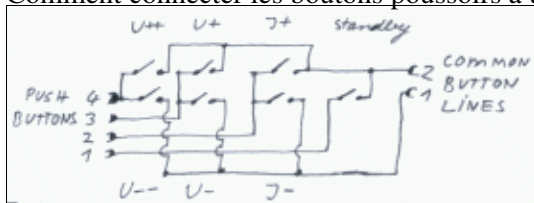
Le schéma principal (cliquez dessus pour une image plus grande) :



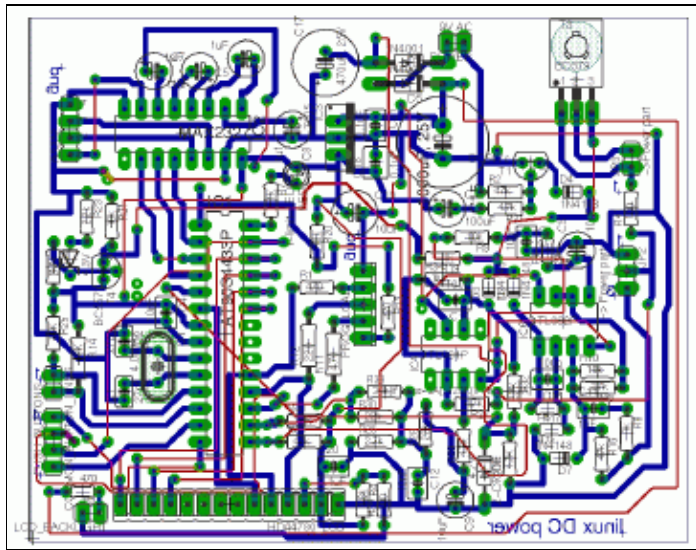
Le schéma pour la partie puissance (cliquez dessus pour une image plus grande) :



Comment connecter les boutons poussoirs à une matrice (cliquez dessus pour avoir une image plus grande) :



Le circuit principal, vue de dessus (cliquez dessus pour une image plus grande) :



Le circuit est spécialement conçu pour les électroniciens amateurs. Seule la couche bleue est destinée à être gravée sous forme de circuit imprimé. Les lignes rouges sont des fils. Une carte simple face est plus facile et demande moins de précision dans sa conception. Vous pouvez disposer les fils (en rouge) de manière à ce qu'ils soient le plus courts possible. Je n'ai pas réussi à le faire avec eagle.

Les quelques composants de la partie puissance de l'alimentation peuvent être montés sur des cartes prototypes standard (ces cartes avec de nombreux trous). La carte principale et la partie puissance sont connectées par des fils (JP2 et JP3). Vous noterez que le fil de masse de la partie principale se connecte à la sortie de courant. C'est correct et c'est la raison pour laquelle nous avons besoin de deux transformateurs (un pour la partie puissance et l'autre pour la partie logique du micro-contrôleur et des amplificateurs).

## Comment ça marche

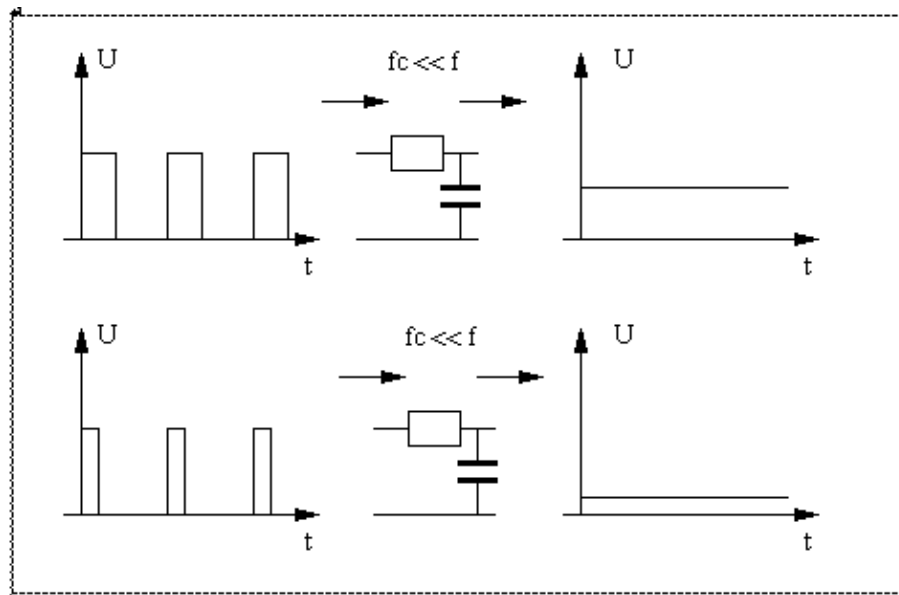
En regardant le schéma principal, vous pouvez constater qu'il est constitué de deux blocs logiques. L'un est marqué comme « contrôle de courant » et l'autre comme « contrôle de tension ». Ce sont deux boucles de contrôle indépendantes. Une boucle contrôle la tension de sortie pendant que l'autre surveille la chute de tension de la résistance de 0.275 Ohm dans la partie puissance. La perte de tension est équivalente au courant. Les deux blocs de contrôle sont « combinés » via les diodes D2 et D3. Ces diodes forment une porte OR analogique. C'est-à-dire que si le courant est trop élevé, la partie de contrôle de courant abaisse la tension jusqu'à ce qu'elle soit au-dessous de la limite (courant suffisamment bas) et la partie de contrôle de tension est chargée de réguler la tension en sortie.

Ce OU (OR) logique fonctionne car le transistor T3 est connecté par R19 au +5V. S'il n'y avait pas d'amplificateurs opérationnels connectés derrière D2 et D3 vous obtiendriez le maximum de puissance de sortie. Les amplificateurs opérationnels dans la boucle contrôlent la sortie en enlevant le +5V de T3 (il en détourne autant que nécessaire vers la masse).

La boucle de contrôle de tension asservit la tension de sortie en fonction du niveau obtenu sur la broche 5 de IC6B. En d'autres termes, la tension sur la broche 5 est équivalente à la sortie multipliée par le facteur d'amplification qui est déterminé par les résistances R15, R10 et R16. La même chose se produit pour le courant sauf qu'il s'agit de la tension sur la résistance R30 et qu'elle est équivalente au courant de sortie maximum.

De manière à obtenir le courant maximum ou à réguler la sortie de l'alimentation, nous devons seulement

fournir les tensions appropriées sur les deux points (broche 5 d'IC6B et résistance R30). C'est ce que fait le micro-contrôleur... mais comment peut-il générer et réguler une tension de référence ? Regardez l'image suivante :



Ce que vous voyez sur l'image est la manière dont un signal par impulsion peut être transformé en un signal continu. Il suffit de le faire fonctionner au travers d'un filtre passe-bas avec une fréquence de coupure cent fois (ou plus) inférieure à la fréquence du signal. Comme notre micro-contrôleur fonctionne à 4 MHz, il n'est pas difficile de concevoir un tel filtre passe-bas. Même si nous mettons en place la génération du signal par logiciel, nous obtiendrons encore quelques kHz et le filtre sera encore très petit.

La différence dans l'image entre les deux schémas est appelée modulation par impulsion. En changeant la longueur de l'impulsion, nous pouvons modifier la tension derrière le filtre.

C'est bien, non ? Nous pouvons générer une tension exacte à partir d'un signal numérique !

Le micro-contrôleur AT90S4433 possède deux compteurs internes. L'un est de 16 bits et l'autre de 8 bits. Le compteur 16 bits peut utiliser la modulation à largeur d'impulsion (PWM) qui existe déjà matériellement dans le circuit AT90S4433 avec une résolution de 10 bits. Le compteur 8 bits n'a pas cette faculté mais nous pouvons la mettre en place par logiciel. Il est encore suffisamment rapide. Nous utilisons le compteur 16 bits pour la régulation de tension, ce qui nous donne 10 bits = 1023 pas de résolution pour le contrôle de tension. Le courant de sortie est contrôlé par le compteur 8 bits, ce qui nous donne 255 pas pour contrôler 1–3000 mA. Cela signifie que nous avons une précision de près de 12 mA (ou moins). C'est encore suffisant pour le contrôle du courant.

Toutes les autres parties du circuit concernent l'alimentation et la tension de référence (le 7805 est notre point de référence) et permettent de s'assurer que l'alimentation ne devient pas instable lors de la mise sous tension ou hors tension.

## Le logiciel

Le logiciel pour le micro-contrôleur utilise beaucoup d'aspects que vous connaissez déjà depuis les premiers articles (uart pour RS232, afficheur LCD, compteur en mode interruption). Vous pouvez y jeter un coup d'oeil ici :

linuxdcp.c.

Le plus intéressant, est probablement le logiciel PWM (Pulse Width Modulation). La variable `ipwm_phase` gère, avec `ipwm_h`, le PWM pour le courant. Nous lançons simplement le compteur 8 bits en mode interruption et à chaque fois qu'il génère un dépassement, la fonction « `SIGNAL(SIG_OVERFLOW0)` » est appelée. Nous contrôlons ici `ipwm_phase` pour voir si nous devons générer un 1 ou un zéro sur la sortie et nous redémarrons le compteur de temps. Facile.

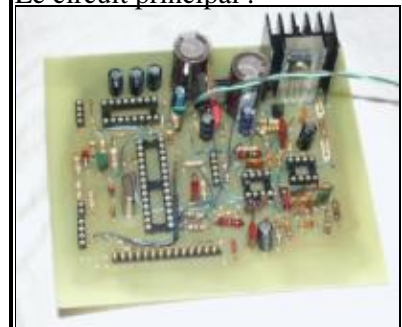
Le logiciel n'est pas du tout compliqué mais pour le comprendre exactement, vous devez lire la documentation technique du 4433 (voyez les références).

Le 4433 est un micro-contrôleur 8 bits et ses capacités mathématiques sont très limitées. Les fonctions `divXbyY` et `multiXbyY` gèrent les maths 24 bits, dont nous avons besoin pour calculer avec précision la largeur de l'impulsion à partir d'une tension donnée, fournie par l'utilisateur.

Notre alimentation possède 7 boutons. 6 boutons sont disponibles pour changer les niveaux de tension et de courant et un bouton sert à la mise en « pause ». En utilisant le bouton « pause », vous pouvez temporairement désactiver l'alimentation et encore changer les limites de tension et de courant. L'état des boutons est « pulled » (relâché) dans la boucle principale du programme. La variable `ignorebutton` est utilisée pour éviter les rebonds des boutons. Lorsque vous pressez un bouton avec le doigt, il rebondit un peu. En tant qu'être humain, nous ne le remarquons pas mais le micro-contrôleur est si rapide qu'il verra un bouton pressé, relâché, pressé... Le compteur `ignorebutton` attend un peu après la pression sur le bouton pour éviter ce rebond.

## Créer le circuit imprimé

Le circuit principal :



Le boîtier pour l'alimentation. Du bois pour les cotés, des feuilles de métal pour le fond, le dessus et la face avant :



La face avant :



Le paquetage logiciel contient un fichier postscript (linuxDCpower.ps) pour le circuit imprimé. Je trouve personnellement que les pastilles sont toujours un peu trop petites. C'est la raison pour laquelle je recommande fortement de les élargir un peu avec un marqueur avant de graver le circuit. Le processus pour réaliser un circuit à la maison est décrit dans : [Mai 2002, Un panneau de contrôle LCD pour votre serveur Linux](#).

Comment construire un boîtier économique mais esthétique pour votre alimentation est décrit dans l'article « [Septembre 2002, Compteur de Fréquence 1Hz–100Mhz avec afficheur LCD et interface RS232](#) ». Vous pouvez voir le boîtier et la face avant que j'ai réalisés sur la droite. Cliquez sur les images pour des images plus grandes.

## Tester

Comme pour tout circuit que vous soudez, il est raisonnable de ne pas connecter directement les éléments à l'alimentation mais de procéder par étapes. Cela permet de détecter les erreurs faites pendant la construction du circuit.

1. Assemblez le circuit principal mais sans mettre les composants dans leurs supports.
2. Prenez une batterie de 9V et connectez le plus à la broche 2 et le moins à la broche 1 du connecteur marqué AC\_POWER sur le schéma. Utilisez un voltmètre et vérifiez que vous avez du +5V sur le max232 entre les broches 8 et 16 et sur les broches 7 et 8 du micro–contrôleur. Sur les amplificateurs opérationnels, vous devez avoir presque 9V sur la broche d'alimentation positive.
3. Branchez maintenant la batterie 9V (broche 1 au plus et broche 2 au moins) et vérifiez que vous avez –9V sur les broches d'alimentation négatives des amplificateurs opérationnels.
4. Si tous les tests sont passés jusque là, alors l'alimentation de la carte principale fonctionne et il n'est plus risqué d'insérer le max 232 et le micro–contrôleur dans leurs supports.
5. Utilisez encore la batterie de 9V et connectez–la de manière à ce que le +5V fonctionne (voir ci–dessus). Connectez le câble du programmeur au port parallèle et au connecteur pour programmer le circuit.

Décompressez le logiciel (pour le téléchargement, voyez le chapitre des références), « cd » dans le répertoire qui est créé et tapez :

```
make avr_led_lcd_test.hex
make testload
make ttydevinit
```

Le logiciel de test doit maintenant être chargé dans la carte. Sur l'afficheur du LCD, vous devez voir « hello », la LED rouge doit clignoter et si vous connectez votre ordinateur à la rs232, vous devriez voir s'afficher « ok » (initialisez la ligne RS232 avec ttydevinit, puis tapez cat /dev/ttyS0 ou cat /dev/ttyS1 pour COM2).

6. Assemblez maintenant la partie alimentation mais ne connectez pas encore le transformateur principal. Connectez plutôt la batterie de 9V aux câbles où le transformateur sera connecté. Quel que soit le sens de connection de la batterie, le condensateur de 4700uF doit toujours être chargé autour de 9V. Contrôlez–le avec un voltmètre.
7. Lorsque la dernière étape de test est passée, faites un contrôle final des fils et connectez ensuite tous les transformateurs et alimentez le tout. Sans amplificateurs opérationnels dans leurs supports, vous devriez obtenir la valeur maximum de tension de sortie. Mesurez–la en faisant attention de ne pas

créer de court circuit sinon vous grillerez les transistors puisqu'il n'y a pas encore de limitation de courant.

- Débranchez, insérez les amplificateurs opérationnels et reconnectez le câble du programmeur, mettez sous tension et tapez :  
make  
make load
- L'alimentation doit être maintenant pleinement fonctionnelle. Notez que pendant que le câble du programmeur est encore connecté, la sortie est toujours désactivée. Débranchez-le pour avoir la bonne tension et le bon courant en sortie.

## Et voici notre propre alimentation

Vous avez vu que nous avons trois options disponibles en fonction du transformateur utilisé. Le logiciel est par défaut pour une tension de sortie de 16V, 2.2A. Pour le modifier, éditez le fichier `linuxdcp.c` et recherchez :

`MAX_U`, `IMINSTEP`, `MAX_I`, et dans la fonction `set_i`, vous devez changer la calibration si vous avez une sortie de 3A maximum. Le code est bien commenté et vous verrez ce que vous devez changer.

Enfin, voici quelques images de l'alimentation telle que je l'ai fabriquée. Cela représente du travail mais c'est vraiment une alimentation très bonne et très robuste. Le temps investi est rentable car une alimentation de laboratoire est réellement une des choses les plus utilisées.





# Utilisation de l'alimentation

La manière d'utiliser l'alimentation est à peu près évidente. Vous avez 4 boutons pour définir la tension de sortie. 2 boutons pour augmenter/diminuer de 1V et 2 boutons pour augmenter/diminuer de 0.1V. La limitation de courant peut être aussi définie par deux boutons. Mais ici, l'échelonnage n'est pas linéaire. Pour des valeurs plus petites, vous pouvez incrémenter ou décrémenter par pas de 50mA. Pour des valeurs supérieures à 200mA, vous avez des pas de 100mA et de 200mA au-dessus de 1A. De cette manière, il est facile de parcourir l'échelle complète avec seulement deux boutons.

Le bouton attente peut être utilisé pour désactiver temporairement l'alimentation sans avoir besoin de redéfinir les valeurs lors du retour sous tension.

La LED rouge s'allumera lorsque vous atteindrez la limite du courant et elle clignotera en mode attente.

L'alimentation peut aussi être totalement contrôlée par des commandes ASCII via la ligne série RS232. Les commandes suivantes sont disponibles :

u=X définit la tension (i.e. u=105 fixe la tension à 10.5V)

i=Xmax détermine le courant maximum (i.e. i=500 définit la limite de courant à 500mA)

s=1 ou s=0 met en mode attente

u=? ou i=? ou s=? affiche les valeurs actuelles. Par exemple :

u: 50 s:0 i: 100 l:0

u: signifie tension = 50 = 5V, s:0 signifie mode attente désactivé, i: 100 représente 100mA et l:0 signifie que la limite de courant n'a pas été atteinte.

En utilisant ce langage de commande ASCII, vous pouvez aussi écrire une interface de commande graphique pour l'alimentation. Pour utiliser la ligne RS232, vous devez d'abord l'initialiser par la commande `ttydevinit`, qui est incluse dans le paquetage logiciel. Tout ceci est décrit dans l'article de [Septembre 2002, Compteur de fréquence](#).

Comme vous l'avez vu dans le schéma au-dessus, nous utilisons deux transformateurs et le plan de masse de la logique de contrôle est connecté à la sortie positive. Les deux transformateurs séparent les tensions et il n'y a normalement pas de problème avec ce procédé. Nous devons établir les connexions de cette manière pour avoir la bonne polarité sur la boucle de retour des amplificateurs opérationnels. **Un mot d'avertissement :** Cette installation signifie aussi que la ligne de terre de la ligne RS232 est connectée à la sortie positive ! En d'autres termes, vous ne pouvez pas utiliser la ligne RS232 si vous utilisez l'alimentation avec des éléments qui sont connectés par ailleurs à la masse de votre ordinateur. Il pourrait être judicieux de mettre une étiquette sur le boîtier de l'alimentation indiquant « la ligne de masse de la connexion RS232 est connectée à la sortie positive ». Si vous voulez vous assurer qu'il n'y a pas de possibilités de court circuit avec les fils de masse de la RS232, utilisez alors une batterie de portable ou vérifiez que le circuit alimenté ne possède pas d'autres connexions ou n'utilise pas l'interface de commande RS232. Ne soyez pas trop inquiets de cette mise en garde. Si vous ne dépassez pas 250mA sur l'alimentation, la LED rouge vous indiquera que vous avez fait une erreur et qu'il n'y a pas de danger pour votre ordinateur même si vous avez fait quelque chose de stupide.

## Sécurité

Ce circuit contient un transformateur qui est connecté à l'alimentation principale (230V ou 110V en fonction de votre pays). Veuillez vous assurer d'une bonne isolation. Si vous n'avez jamais travaillé avec des alimentations alors demandez à des personnes expérimentées de contrôler votre circuit pour ce qui concerne l'isolation et la sécurité avant de brancher pour la première fois.

# Ajustement

Le logiciel pour l'alimentation est déjà calibré. Dans la plupart des cas, il n'y aura rien à modifier. Le matériel concerné par la calibration dépend seulement des 7805, R15, R10, R16 et R38, R30, R26. Seuls ces composants influencent la tension et les niveaux de courant. Si vous voulez faire un ajustement plus fin, vous pouvez soit changer ces résistances, soit modifier le logiciel. Notez qu'un câble de programmeur connecté influence la sortie. Avant de faire des mesures, vous devez déconnecter le câble. Dans le logiciel, vous pouvez faire des modifications dans les fonctions `set_u` et `set_i`. Ceci est commenté dans le code de [linuxdcp.c](#)

## Références

- Le logiciel du programmeur uisp AVR : [www.amelek.gda.pl/avr/](http://www.amelek.gda.pl/avr/)  
copie locale : [uisp-20011025.tar.gz](#)
- Comment construire le programmeur et installer le compilateur AVR :  
[Mars 2002, Programmer le micro-contrôleur AVR avec GCC](#)
- Le code source pour cet article [linuxdcpower-0.1.tar.gz, 1201K](#) . Le schéma, les fichiers Eagle et les captures d'écran sont aussi incluses.
- Tous les logiciels (les mises à jour seront indiquées ici) et leur documentation : [software/datasheets](#)
- Fiche technique pour le BD379 [bd379.pdf 44K](#)
- Fiche technique pour le TL082 [TL082.pdf 110K](#)
- Fiche technique pour le TL071 [TL071.pdf 268K](#)
- Fiche technique pour le 2N3055 [2n3055.pdf 64K](#)
- Fiche technique pour le MAX232 [MAX220-MAX249.pdf 448K](#)
- Fiche technique pour le ST232, une variante bon marché, souvent vendue à la place du vrai MAX232 [st232.pdf 100K](#)
- Fiche technique pour l'Atmel AT90S4433 [avr4433.pdf 2356K](#)
- Le site web d'Atmel : [www.atmel.com/](http://www.atmel.com/)
- Eagle pour Linux [cadsoftusa.com](http://cadsoftusa.com)

Site Web maintenu par l'équipe d'édition LinuxFocus  
© Guido Socher  
"some rights reserved" see [linuxfocus.org/license/](http://linuxfocus.org/license/)  
<http://www.LinuxFocus.org>

Translation information:

en --> -- : Guido Socher ([homepage](#))

en --> fr: Iznogood <[iznogood@iznogood-factory.org](mailto:iznogood@iznogood-factory.org)>